# Page and page scan

**Overview**

Page 过程主要为两个蓝牙设备的连接做准备。如果两个蓝牙设备的时钟同步，跳频序列一致，以及传输数据包的Access Code一致，可以认为两个设备已经连接。对于连接的两个蓝牙设备，有一个主设备Master，和一个从设备Slave（我们可以认为Page Device为Master，Page Scan Device为Slave）。他们的物理通道由Master决定。这是因为两设备的跳频序列选择和Master设备的BD\_ADDR以及时钟有关。 Slave若要传输数据给Master，需要知道Master的时钟和地址信息。因此，要想进行两个蓝牙设备的连接通信，需要把Master的信息（包括LAP，UAP，LT\_ADDR，CLK等）通过发送FHS包给Slave，这样Slave可以根据FHS包里的信息进行跳频，与Master进行连接传输数据。

在两个蓝牙设备进行Page之前，Master可以通过inquiry或者之前的连接，获取到Slave的蓝牙地址（BD\_ADDR）等信息，也可以对Slave的时钟进行预估（这个估计是不准的，可能会加快page过程）。对于Slave来说，他只知道自己的信息。

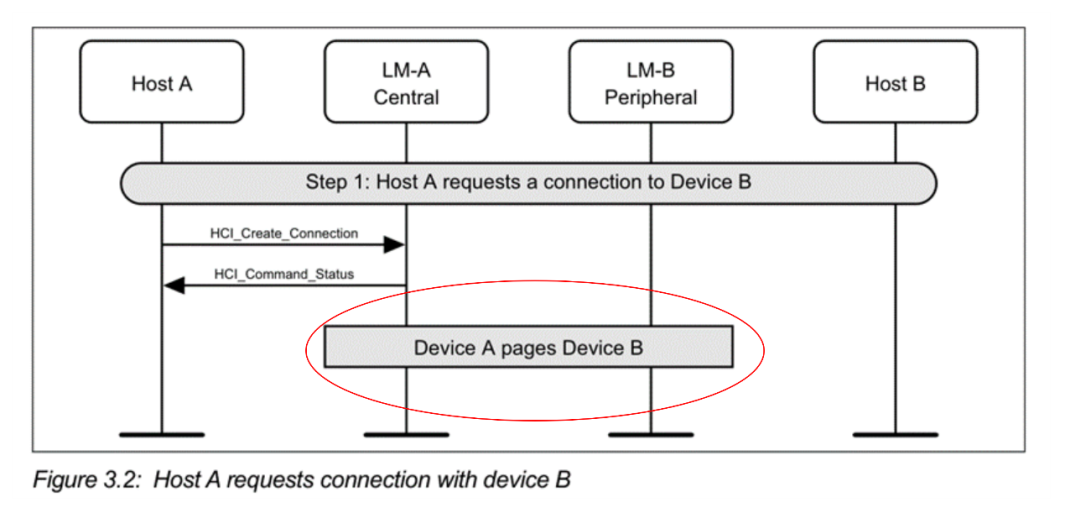
在这种情况下，Master需要利用Slave的信息（主要是地址信息）与Slave建立稳定链接。

因此，整个page过程（包括Page, Page Scan, Central Response, and Peripheral Response substates），数据包的发送和接收，跳频的选择是根据Slave的BD\_ADDR 信息进行的。数据包的Access Code来自于paged device’s BD\_ADDR。

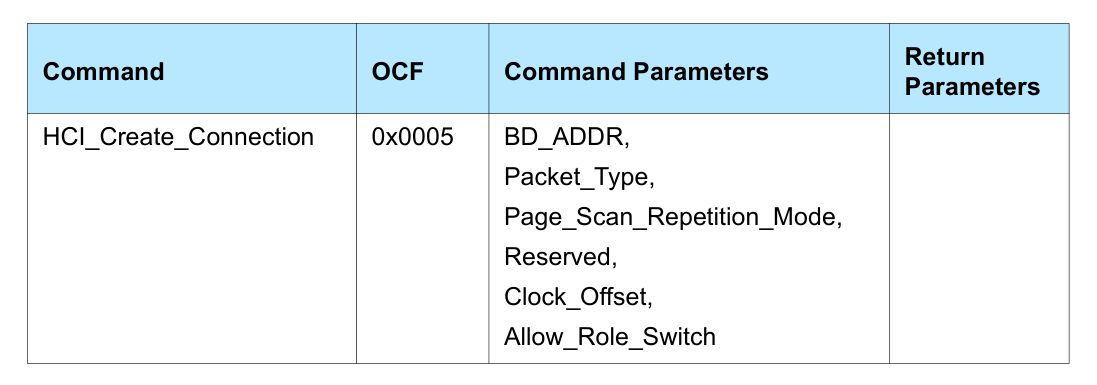
## The page procedure

当两个设备需要建立连接时，主设备的Host向controller发送一个HCI\_Create\_Connection命令。然后，controller使用指定的从设备的BD\_ADDR执行基带寻呼过程，从设备的BD\_ADDR从inquiry过程获取得到。

两个通道时间，频率和接入码都保持一致的情况下，认为两个通道是连接的。由于page过程是在为设备连接做准备，因此该过程主要是为了收集需要建立连接的两个设备间的地址，时钟信息等。

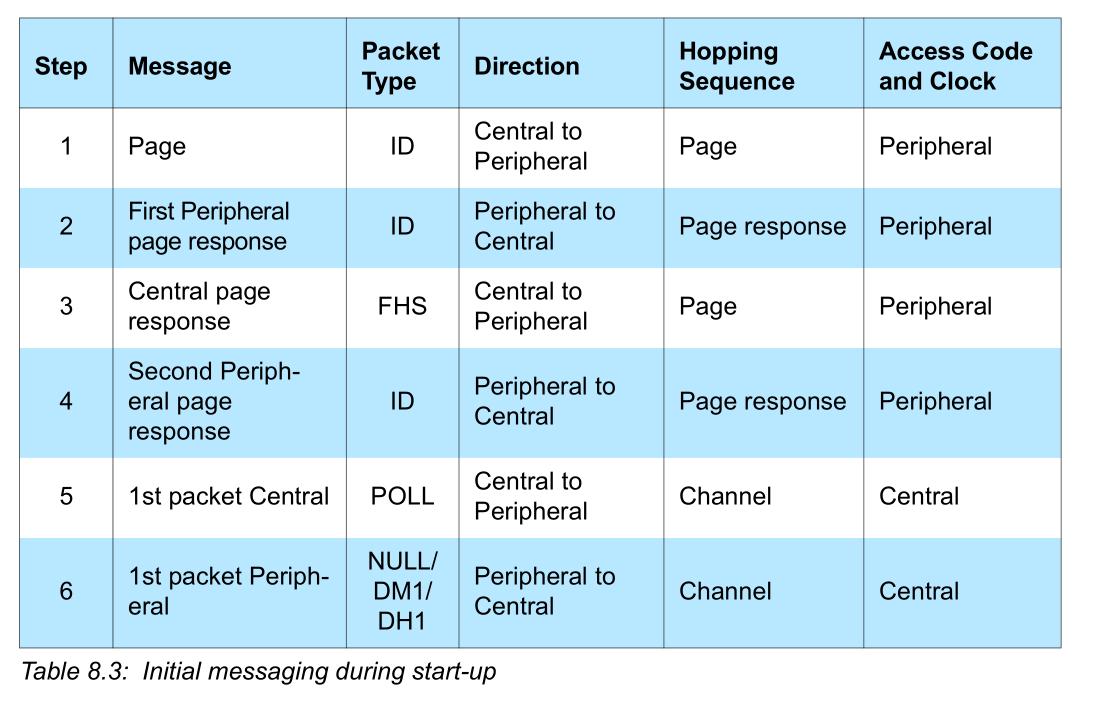


HCI\_Create\_Connection命令：



寻呼过程按照如下步骤进行：

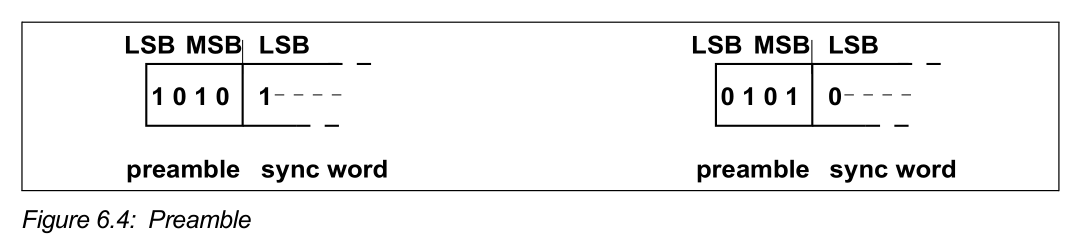
1. 一个设备（源）发送含有设备接入码的ID包寻呼另外一个设备（目的），此时处于寻呼状态(Page state)。
2. 目的设备接收到该寻呼消息，此时处于寻呼扫描状态(Page Scan state)。
3. 目的设备向源设备发送含有设备接入码的ID包的响应，此时处于子设备响应状态(Peripheral Response state)。
4. 源设备发送FHS包到目的设备，此时处于主设备响应状态(PageResponse state)。



## Packets

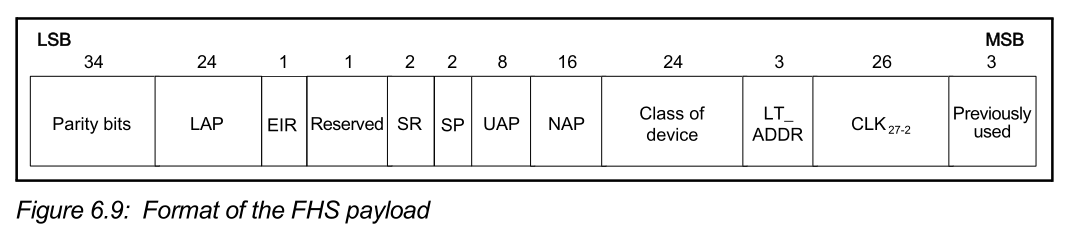
* ID包

ID包由从设备接入码DAC (device access code)组成，它的长度固定为68位，由4位preamble和64位sysc word构成。其中sysc word的64bit主要来源于从设备的低24bit地址LAP。



* FHS包

FHS包是一个特殊的控制包，其中包含蓝牙设备地址和发送者的时钟。有效载荷包含144个信息位加上一个16位的CRC码。



* NULL包

NULL包没有有效载荷，仅由通道接入码和报头组成。它的总(固定)长度是126位，可以用来向源返回关于前一个传输(ARQN)成功的链路信息，或者RX缓冲区(FLOW)的状态，NULL包不需要确认。

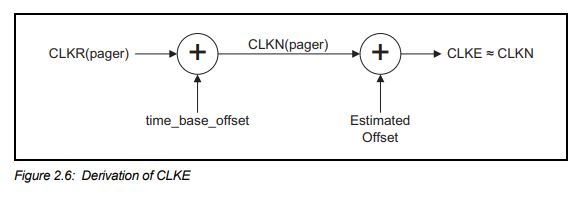
* POLL包

POLL包与NULL包非常相似，它没有有效载荷。与NULL包相比，它需要接收方的确认。

## Bluetooth Clock

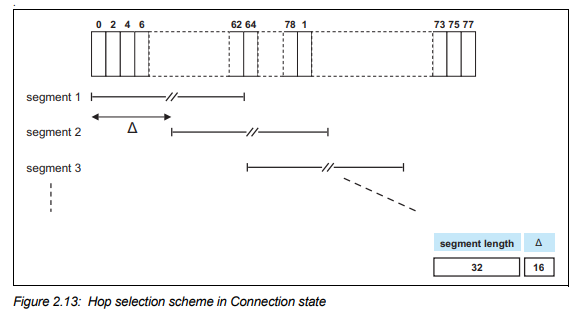
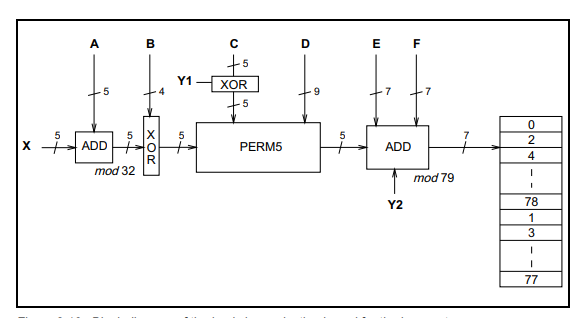
图示

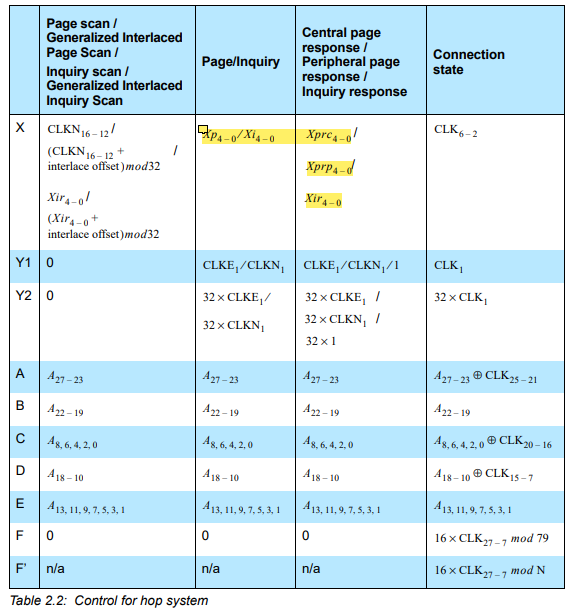
中度可信度描述已自动生成

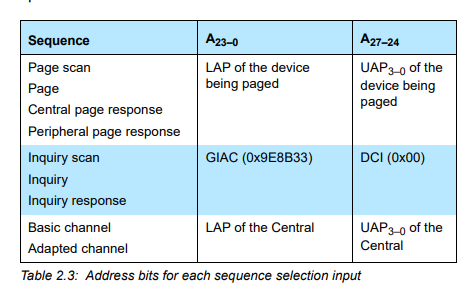


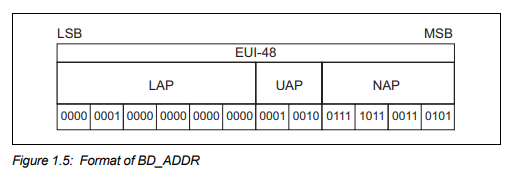
蓝牙时钟CLK，28位的计数器，最低位CLK0计数一次是312.5us，CLK1对应一个slot。Page和Inquiry的跳频是根据CLK0计数，在TX slot内每秒3200次跳频。

## Hop Selection





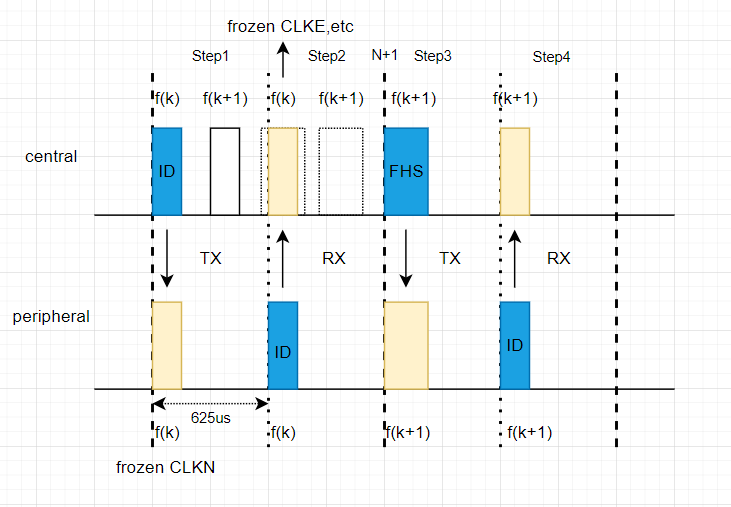




跳频选择算法将经典蓝牙上79个频点的选择按照一定的顺序进行排列选择，按照顺序偶数在上，奇数在下。跳频算法将按顺序排列的频点分段，每段包含32的频点。

参数ABCDEF可以确定跳频选择发生在哪一个Segment，X用来确定选择的跳频点在Segment里的具体位置。

Page整个过程，参数ABCDE的值都取自Slave的BD\_ADDR，因此，Master和Slave的频率选择都发生在同一个Segment。

Page hopping sequence：

手机屏幕截图

中度可信度描述已自动生成

Peripheral page response hopping sequence：

图片包含 徽标

描述已自动生成

Central page response hopping sequence：

文本

中度可信度描述已自动生成

*关于Page过程Central设备与Peripheral设备使用跳频选择：*

Step1中central发出message的频率和Step2中等待接收的频率一致，都在同一个trains序列；对于Peripheral来说，当他以一个频点（这个频点在central跳频trains序列内，可能是A-train或者是B-train）接收到Page message的时候，他就会知道，在625us之后，Step2，central以此时相对应的频点来接收Peripheral响应信息，所以peripheral会冻结此时的时钟信息（即冻结此时的频点，过625us后再次使用）。与此同时，冻结的时钟信息也会用于后面的接收和响应。同样，当central接收到Peripheral响应的时候，Central也会冻结此时的CLKE等信息。此时可以认为Central和Peripheral使用冻结的时钟信息建立同步了，并用于后面状态的信息交互。而且每进入到一个TX，Central和Peripheral的跳频同时加一。后面的状态中，两个设备的频率的选择就和A、B-train序列无关了，他们就会在已经确定了的segment中的32个跳频序列同步跳频，直到两个设备建立链接。

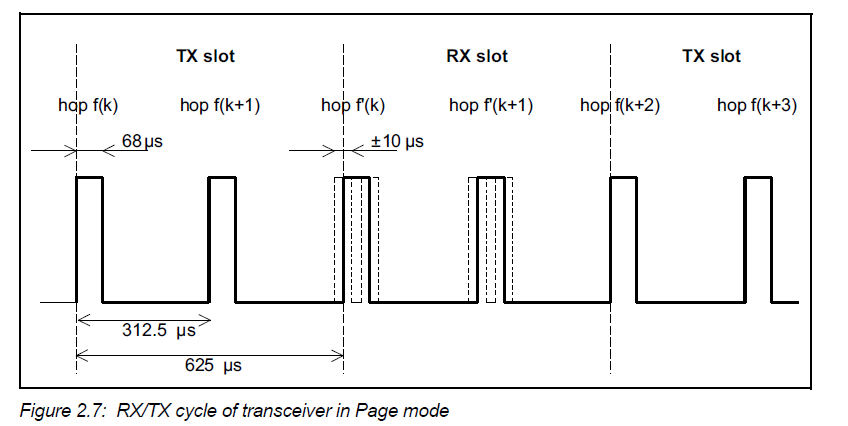
***对在page进程中，Step1-4中主从设备的跳频选择，做出以下更新：***

根据跳频选择公式，对跳频影响的不仅有参数X的影响，还应考虑Y1和Y2对跳频选择的影响，Page Scan中，Y1和Y2均为0，而Page和Central/Peripheral Page Response的Y1和Y2不为0，和CLKE1和CLKN1有关，在发送和接收状态时分别为0或1，会对跳频的计算产生影响，使得f(K)和f’(k)并不是相等的。虽然两者不是相等的，但是两者是相互对应的，可以正常进行后面步骤，直至建立连接。

## Timing

* **Paging procedure timing**

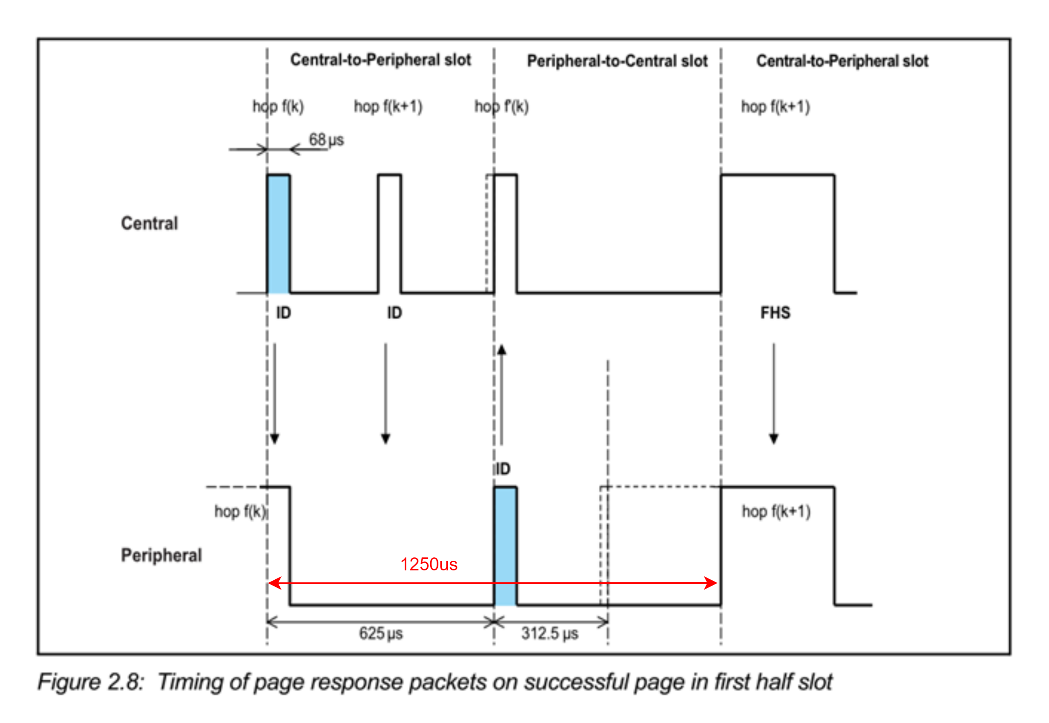
由于page消息是一个非常短的包，因此page过程使用的跳率为3200跳/s（312.5us更新一次频率）。在单个TX槽间隔内，寻呼设备应在两个不同的跳频率上传输。第一次传输从CLK0 = 0开始，第二次传输从CLK0 = 1开始。在一个单一的RX槽间隔中，page设备将在两个不同的跳频上侦听外设页面响应消息。与传输类似，接收在CLK0 = 0处开始，第二个接收在CLK0 = 1处开始。



* **Page response timing**

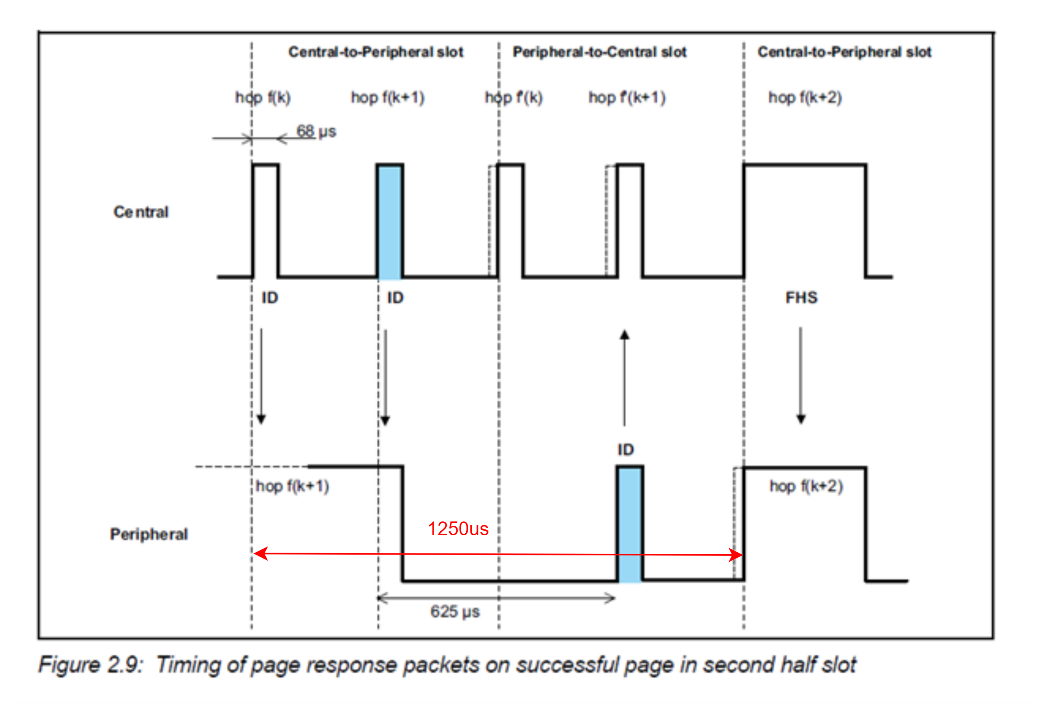
第一个半槽page消息成功发送时的过程：

第一个Peripheral page response的发送时间与收到page消息的包的间隔为625us，第一个Central page response的发送时间与第一个page消息的间隔为1250us。



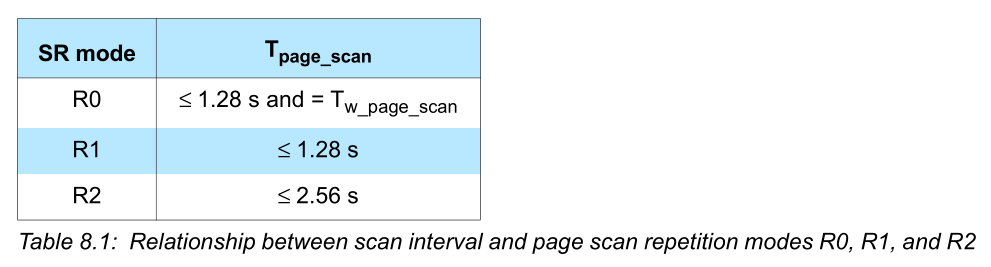
第二个半槽page消息成功发送时的过程：

第一个Peripheral page response的发送时间与收到page消息的包的间隔为625us，第一个Central page response的发送时间与第一个page消息的间隔为1250us。

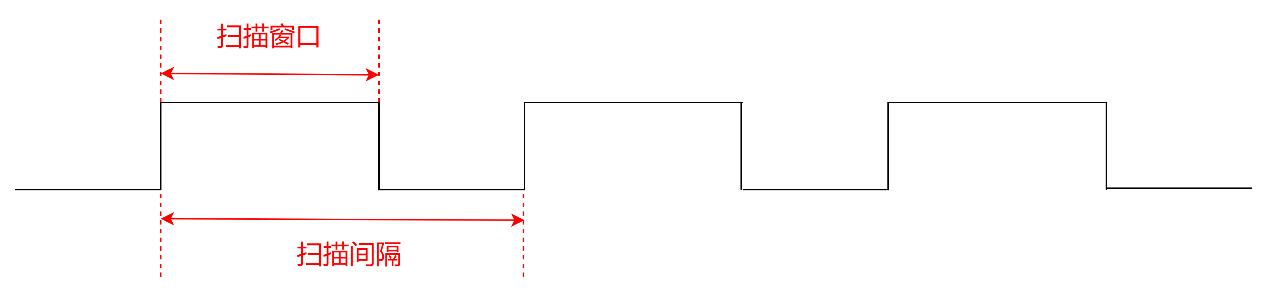


## Page scan substate

Page scan通道的跳频相位由扫描设备的本地蓝牙时钟CLKN16-12决定，因此该通道1.28s进行一次跳频。跳频序列由扫描设备的蓝牙地址决定。扫描可进行两种设置：标准扫描和通用交错扫描。在标准扫描状态下，设备扫描窗口Tw\_page\_scan默认持续11.25ms，在窗口持续时间内，扫描频率保持不变。扫描间隔Tpage\_scan表示两个扫描开始时间间隔，根据时间长短可分为下图三个模式：



扫描窗口与扫描间隔之间的关系示意图：

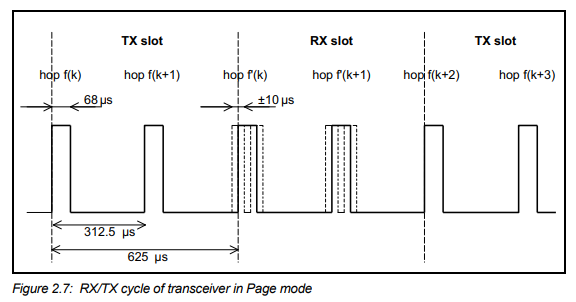


## Page substate

由于Master和Slave的时钟没有同步，Master并不清楚在什么时候，用多少频率与slave进行交互，因此，Master会在不同的跳频点发送Page message，并监听Slave的响应。

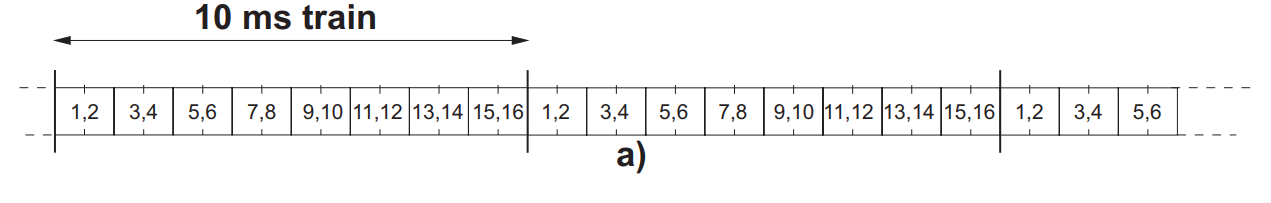
Master发送Page Message时，Host首先要向Controller发送Slave的BD\_ADDR，根据Slave的蓝牙地址中的LAP和UAP，进行跳频序列选择（选择segment）。

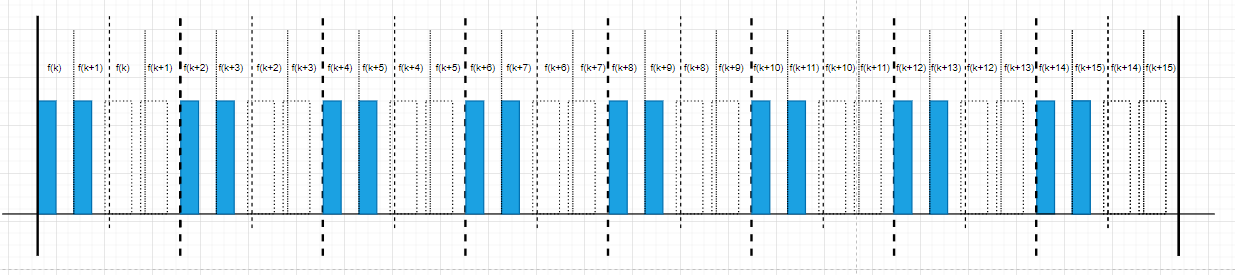
跳频频点的选择主要跟参数CLKE4-2,0有关。因为用到CLK0，所以两个跳频之间间隔半个slot，



Master在TX slot发送两个不同频率的Page message，在RX slot处对应的频率监听Slave响应信息。

按照上面的跳频计算公式，CLKE4-2,0变化一个周期，时钟计数是25，即16个slot，10ms，由于没有用到CLKE1，所以一个周期只能监听16个频点。

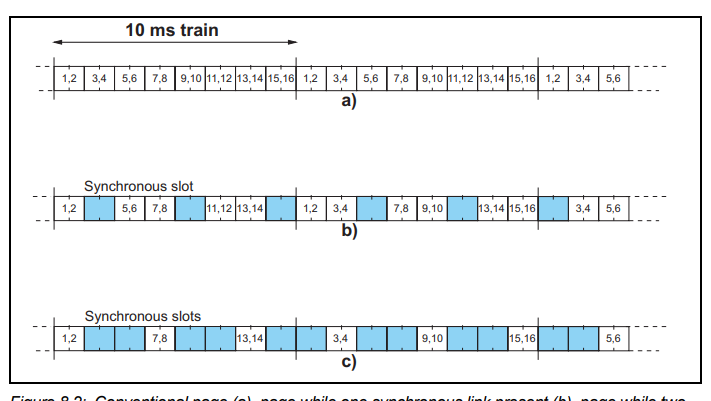




Page Scan设备扫描的频率在segment的位置由X=CLKN16-12确定，每经过1.28s，扫描设备会换一个频率进行扫描。

主设备大多情况下不能很精确的预测Slave的时钟，会有一些偏差，如果时钟偏差在-8x1.28 s 到 +7x1.28 s以内，即预测的Slave扫描频率为f（k），实际扫描频率在f(K-8)—f(k+7)这个范围内（即A-train），那么用这16个跳频序列循环重复Npage次发出page message，扫描设备会在某一时刻收到page信息。如果实际的时钟偏差小于-8x1.28 s，或者大于+7x1.28 s，则有可能扫描频率不在f(K-8)—f(k+7)这个范围内，而在另外16个跳频序列中f(k-16), f(k-15),...,f(k-9),f(k+8)...,f(k+15)，即B-train。通过改变offset值，可以改变A-train和B-train。A-train和B-train交替使用，直到Master接收到响应或者发生超时。

如果有同步链接的话，由于SCO/eSCO的优先级高，所以Page过程会被中断，会为SCO/eSCO提供传输，此时需要考虑扫描窗口的大小，以保证能够完整的扫描16个跳频序列。扫描周期的长度取决于存在的同步保留时隙。



情况a），不存在同步传输，所以，10ms可以扫描全部16个频率点，考虑扫描窗口可能存在偏移，最小的Tw\_page\_scan=10.625ms，一般默认是11.25ms；

情况b），存在一个SCO传输，且TSCO=6slots，这样的话就至少需要36个slots的扫描窗口进行扫描，才能扫描全部16个频点。

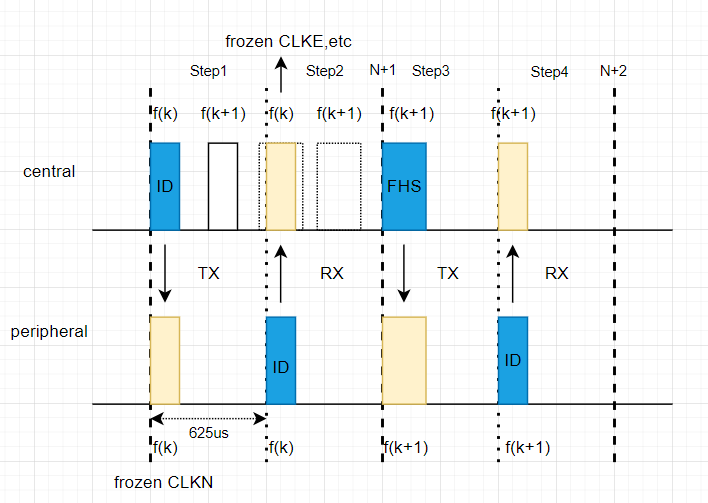
## Page response substates

当peripheral成功接收到page消息后，其实两个设备就已经初步建立同步了。这时候两个设备交换一些重要信息（如Master发送的FHS包），这样两个设备就可以使用同一个channel access code，相同的跳频序列和同步的时钟进行通信。

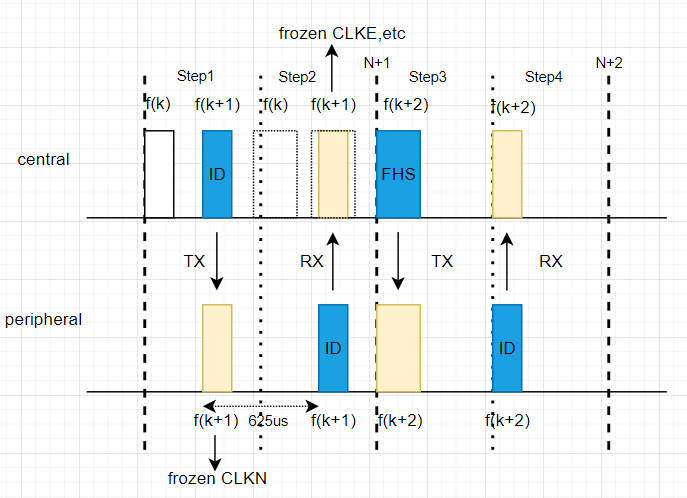
Page 过程主要分为4步

Step1：Master在不同的频率下发送Page信息（ID包），Slave在一个变化较为缓慢的跳频下进行扫描。交替使用train A和train B当从设备接收到page 信息时，进入Step2。

Step2：从设备在接收到page信息后，经过625us，与在step1中接收信息频率相对应的频率想主设备发送响应包（ID包），主设备接收到响应包后，进入Step3。



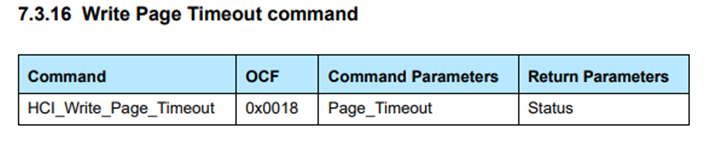
Step3：从设备发送响应包后，等待接收主设备发送FHS包。主设备发送FHS包时，CLK0和CLK1复位为0。如果没有接收到FHS包，每隔1.25ms（CLK1set to 0），用下一个频点去等待接收。如果在*pagerespTO*时间内从设备都没有接收到FHS包，从设备回到扫描状态，并额外扫描一个周期，如果在此额外扫描期间未收到寻呼信息，则外围设备应以其正常扫描间隔恢复扫描；如果从设备收到主设备发送的FHS包，在接收时刻，从设备的CLK1和CLK0都为0，page进程进入Step4。此时使用的跳频序列仍是Page跳频序列。

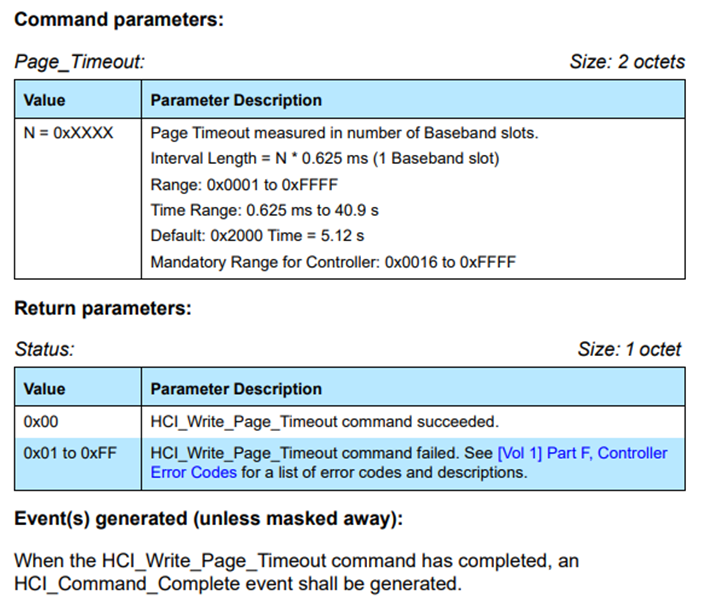


Step4：从设备向主设备发送响应，以确认FHS数据包收到，发送的内容为从设备的device access code。如果没有收到响应，Master会重新传输FHS数据包，每次传输都更新时钟信息，传输过程仍使用从设备的一些参数，直到响应被收到，或者超时。如果在*pagerespTO*时间内主设备都没有接收到响应，主设备会返回Page子状态，并向基带资源管理器报错。如果主设备接收到响应信息，进入step5。

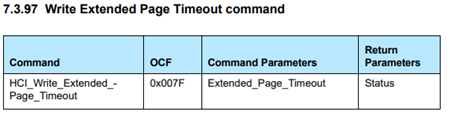
Step5：此时，两设备的时钟同步，使用主设备的Channel Access Code，并使用basic Channel hop sequence。主设备会在TX slot发送POLL包，从设备需要响应主设备，向主设备发送NULL、DM1或者DH1数据包，如果从设备在这一阶段中，经过*newconnectionTO*个slots时间内没有收到POLL包，或者主设备没有收到响应包，他们分别回到Page和Page Scan状态。

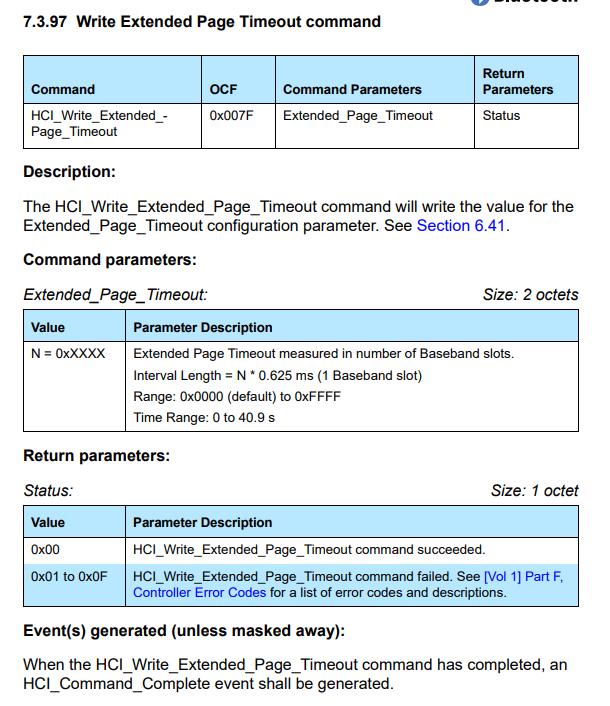
pageTO：当extended\_ pageTO为零时，pageTO定义了在主设备接收响应之前page子状态可以持续的slot数，持续时间为pageTO\*0.625ms。定时器值可由主机改变。HCI提供更改定时器值的命令。



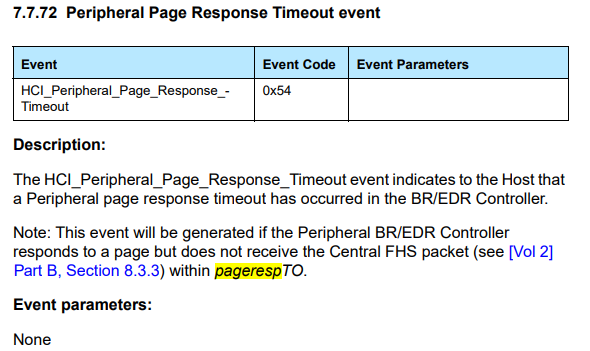


extended\_pageTO：extended\_pageTO定义了在接收到响应之前，page子状态可以持续超过pageTO的额外slot的数量，持续时间为（pageTO + extended\_pageTO）\*0.625ms。定时器值可由主机改变。HCI提供更改定时器值的命令。





pagerespTO：在外围设备中，pagerespTO定义了在发送page确认ID数据包后，外围设备应等待中心响应FHS数据包的slot数。在Central中，pagerespTO定义了Central在返回到page子状态之前应等待FHS数据包确认的slot数。中央和外围设备都应使用相同的超时值，以在达到pagerespTO后允许通用的Page/Page Scan间隔。pagerespTO = 8 slots。



newconnectionTO：每次通过Page、Page Scan或Role Switch启动新连接时，中心都会发送一个POLL数据包作为新连接中的第一个数据包。该POLL数据包的传输和确认被用于确认新连接。如果外围设备没有收到POLL数据包，或者中央设备没有收到响应数据包，则中央设备和外围设备都应返回到前一个子状态。newconnectionTO = 32 slots。